NeRFrac: Neural Radiance Fields through Refractive Surface

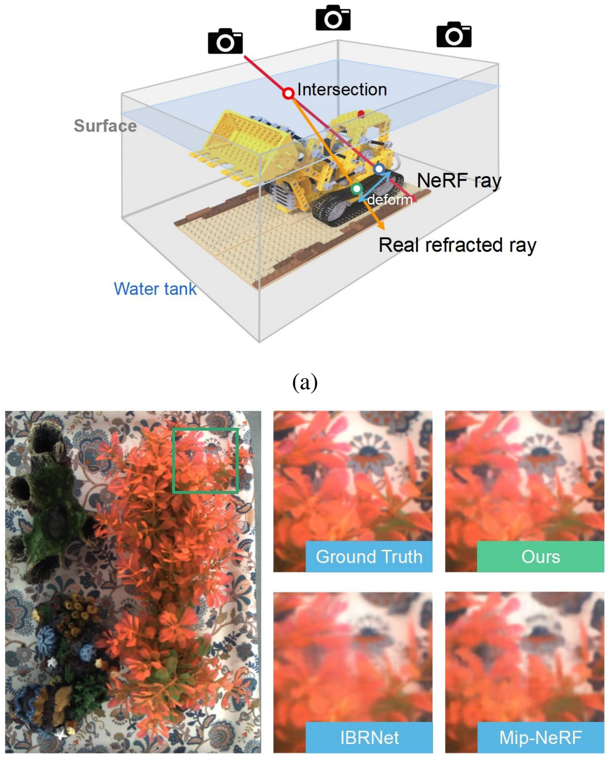
Yifan Zhan* Shohei Nobuhara† Ko Nishino† Yinqiang Zheng*
*The University of Tokyo, Japan †Kyoto University, Japan
zhan-yifan@g.ecc.u-tokyo.ac.jp, {nob, kon}@i.kyoto-u.ac.jp, yqzheng@ai.u-tokyo.ac.jp

Abstract

Neural Radiance Fields (NeRF) is a popular neural representation for novel view synthesis. By querying spatial points and view directions, a multilayer perceptron (MLP) can be trained to output the volume density and radiance along a ray, which lets us render novel views of the scene. The original NeRF and its recent variants, however, are limited to opaque scenes dominated with diffuse reflection surfaces and cannot handle complex refractive surfaces well. We introduce NeRFrac to realize neural novel view synthesis of scenes captured through refractive surfaces, typically water surfaces. For each queried ray, an MLP-based Refractive Field is trained to estimate the distance from the ray origin to the refractive surface. A refracted ray at each intersection point is then computed by Snell's Law, given the input ray and the approximated local normal. Points of the scene are sampled along the refracted ray and are sent to a Radiance Field for further radiance estimation. We show that from a sparse set of images, our model achieves accurate novel view synthesis of the scene underneath the refractive surface and simultaneously reconstructs the refractive surface. We evaluate the effectiveness of our method with synthetic and real scenes seen through water surfaces. Experimental results demonstrate the accuracy of NeRFrac for modeling scenes seen through wavy refractive surfaces. Github page: https://github.com/Yifever20002/NeRFrac.

1. Introduction

Neural view synthesis has become one of the most popular topics in computer vision and graphics, thanks to the breakthrough brought by NeRF[25]. Many follow-up works have extended NeRF in various ways. Parameterization of the viewing direction, however, has basically stayed the same, which fundamentally limits their application to opaque and mostly Lambertian surfaces which is easier to interpolate with a neural representation. Significant departures from this fundamental requirement implicitly imposed by the viewing parameterization can cause dramatic accuracy drop. To handle specular reflection, for instance, a non-trivial reparameterization becomes essential [45].



(a)

(b)

Figure 1: (a) Refraction causes significant erroneous deformations to NeRF as it fundamentally relies on straight-line sampling. It produces artifacts when dealing with scenes behind refractive surface. (b) Novel view synthesis of an underwater scene. Our method is physically-based and renders scenes through the refractive surface accurately in comparison to past state-of-the-art NeRFs.

Another limitation of NeRF is that it samples points on straight rays. This is a stronger assumption than one may think, as it implies that, under any circumstances, light travels along a straight line from the scene surfaces to the viewer (human eyes or cameras). A typical physical phenomenon we encounter in everyday life that violates this assumption is refraction. Consider looking at an underwater scene from outside of the water, for instance, fish swimming in a pond. When viewed from different angles, this refracted underwater scene no longer conforms to the straight ray sampling assumption of NeRF. It would cause unwanted deformations to the scene.

Some NeRFs model light bending with an additional deform layer[31, 33, 13], which outputs an offset of each 3D point to smooth the rendering results. This offset, however, is not physically grounded and cannot model light refraction. We also experimentally find that in underwater scenes, images rendered with a deform layer do maintain visual coherence, but severely deviate from the ground truth.

In this work, we propose NeRFrac, a NeRF that models deformation caused by refraction from its first principles. Instead of simply deforming points in 3D space, we first estimate the depth of the refractive surface and then bend rays instead of points according to the Snell's Law. Our MLP-based neural Refractive Field allows direct inference of intersections between rays and the refractive surface and can implicitly learn the multi-view consistency. Once the intersection is estimated, we can proceed to calculate each refracted ray by Snell's law. Then 3D points are sampled along the refracted ray which are input to our underwater Radiance Field.

Water surfaces are the most commonly encountered refractive surfaces in daily life. Thus, we evaluate the effectiveness of our method on modeling underwater scenes. We build a 3×3 camera array to obtain a novel dataset of real underwater scenes. Objects are placed under the water surface, while the camera array captures images from outside the water surface. To evaluate the accuracy of the simultaneously recovered water surface itself, we also build a novel synthetic dataset using ray tracing [20] to create various complex water surfaces. The experimental results show that our NeRFrac outperforms the related NeRF methods. We also show that by removing the learned Refractive Field, we can view the underwater scene as if it were in air (or the viewer were in the water), just with images captured outside from the water surface. To summarize, we make the following main contributions in this paper:

1) NeRFrac, an end-to-end method for novel view synthesis of scenes captured through refractive surfaces trained with only sparse images;

2) A novel Refractive Field as part of NeRFrac, which explicitly recovers the 3D complex refractive surfaces, whose removal realizes elimination of the refractive surface (*e.g.*, in-water viewing);

2. Related Works

Neural Radiance Field. NeRF[25] is an end-to-end model which represents 3D scenes based on an implicit representation encoded by an MLP. Training of a NeRF is rather simple given multi-view images and corresponding camera parameters. Due to its simplicity in training and high-quality rendering results, NeRF has attracted intensive attention leading to large strides in various directions, including efficiency, accuracy, and complexity of scenes.

FastNeRF[14] divides the MLP into two blocks by factorizing volume rendering, which results in 3000x acceleration. KiloNeRF[37] speeds up the training with thousands of small MLPs, each representing a portion of the scene. Some voxel-based improvements have led to NeRF variants[22, 57, 42, 17, 46, 56] that can be trained very fast. Point-NeRF[53] builds per-point features from dense point clouds with multi-view stereo which enables the use of a smaller and faster MLP for rendering.

Other methods improve rendering with specific designs. Ref-NeRF[45] reparameterizes the view direction to better model specular scenes. NeRFReN[15] proposes to split a scene into transmitted and reflected components to deal with complex reflections. Mip-NeRF[4] replaces each ray with a conical frustum to anti-alias and deblur. Mip-NeRF 360[5] extends Mip-NeRF to unbounded scenes. Some works[29, 47, 54] realize both surface reconstruction and volume rendering to represent smooth 3D shapes of objects. A light field representation is used in a few works[41, 3] to improve rendering quality. Aug-NeRF[9] introduces adversarial training to NeRF. IBRNet[48] applies a ray transformer and can perform real-time rendering. Tensorf[8] applies tensor factorization to achieve faster and better rendering. A handful of methods [51, 38, 10, 28] use depth guides to optimize the results. Other works[11, 7, 43, 16, 2, 30, 36] focus on the generation of portraits. Despite these extensive advancements of the basic NeRF, refractive scene representation has hardly been explored. Several works[3, 6, 13] do apply a NeRF to refractive scenes but do not capture the physical behavior of light which is essential to accurately represent the scene as we demonstrate. Water-NeRF[40] deals with underwater scenes while the camera is put underneath the surface.

Deformation Problem in Novel View Synthesis. The sampling process in NeRF extracts points from straight rays, which is a restricting assumption. In some cases, the consistency between different views is not accurate and causes deformation. Deformation can be introduced between different views of a dynamic scene. Nerfies[31] and D-NeRF[33] design a temporal deformation field so it can interpolate in both space and time. Relative motion of the camera and the scene can also cause deformation. Deblur-NeRF[23] uses a deformation kernel to obtain optimized rays, which approximates the blurring process. LB-NeRF[13] uses a deformation field to simulate the bending of light through transparent medium. The deformation layers in these studies share the same form: $F(\boldsymbol{x}, a) \rightarrow \Delta \boldsymbol{x}$, where $\boldsymbol{x}$ stands for 3D points, and a denotes other related variables (t in D-NeRF for example). In principle, refractive scenes can be interpreted as deformation of the scene since the light itself is deformed during propagation. Inspired by these studies, we introduce an extra Refractive Field before a regular radiance field to explicitly model refraction. We show that this explicit modeling is much more effective in retaining the structured deformation caused by the refractive surface compared to using deformation layers.

Image based Refractive Surface Reconstruction. Reconstruction of refractive surfaces is challenging as refractive surfaces are usually transparent and the scene behind them become dependent on the viewing direction. Past works [52, 44, 27, 21, 58] recover a mesh model and a surface normal map of the water surface from a monocular video. Stereo reconstruction [1, 26] has also been used for water surface reconstruction. For these works, a texture pattern is put underwater to establish correspondences through refraction. Ricardo[12] uses paired images for static refractive surface reconstruction. Our setting is inspired by [34], which adopts a 3×3 camera array for multi-view reconstruction of dynamic refractive surfaces and underwater scenes. NeReF[50] shares similar goals to our work but requires a pattern or images of the scene without water for loss calculation. In contrast, our NeRFrac requires only multi-view images as input, and a water-free pattern is not necessary. Notably, some advanced neural surface reconstruction works have been proposed, such as IDR[55] and VolSDF[54]. IDR requires masked rendering, which is hard to access in unbounded real data. VolSDF optimizes object surfaces based on a neural signed distance function (SDF) network. As we will demonstrate in Sec. 4 and Sec. 6, however, representing refractive surfaces with a neural SDF results in extra sampling and difficulty in gradient backpropagation.

3. Preliminaries

3.1. Snell's Law

Refraction in nature conforms to the Snell's Law, which dictates how much a light ray "bends" when entering a medium with a different index of refraction (*e.g.*, air into water)

$$n_1 \sin\theta_1 = n_2 \sin\theta_2\,, \tag{1}$$

where n_1,n_2 are the indices of refraction (IOR) of two media, and θ_1, θ_2 are the angles of incidence and refraction, respectively.

In 3D world coordinates, angles of incidence and refraction are inconvenient to describe the ray behavior and instead a vector form of Snell's Law can be used. When $\boldsymbol{I} \in \mathbb{R}^3$ is the incident ray and $\boldsymbol{N} \in \mathbb{R}^3$ is the normal vector, the refracted ray $\boldsymbol{T} \in \mathbb{R}^3$ becomes

$$\boldsymbol{T} = \eta(\boldsymbol{I} + c_1\boldsymbol{N}) - c_2\boldsymbol{N}\,, \tag{2}$$

where $\eta = \frac{n_1}{n_2}$, $c_1 = \boldsymbol{N} \cdot \boldsymbol{I}$, and $c_2 = \sqrt{1 - \eta^2(1 - {c_1}^2)}$. We use Snell's Law as a physical guide in NeRFrac. We also build our synthetic underwater dataset so that it strictly conforms to this physical constraint of light behavior. For real captured data, we assume $n_{air} = 1$ and $n_{water} = 1.33$, and for the synthetic data, the IOR is known.

3.2. NeRF and Volume Rendering Revisited

NeRF[25] represents the radiance field of a scene with an MLP. First, points $\boldsymbol{x} \in \mathbb{R}^3$ are sampled along rays calculated from the camera pose. By querying $\boldsymbol{x}$ in the 3D world, a spatial MLP outputs the volume density $\boldsymbol{\sigma}$ of $\boldsymbol{x}$ and a 256-dimensional feature vector. Then, this feature vector is concatenated with the viewing direction $\boldsymbol{v} \in \mathbb{R}^3$ of the ray, and is sent to the direction MLP for radiance $\boldsymbol{c} = (r, g, b)$ prediction. The color of each ray is computed with volume rendering[19]. The expected color of the ray $\boldsymbol{r}(t) = \boldsymbol{o} + t\boldsymbol{v}$ becomes [25]

$$C(\boldsymbol{r}) = \int_{t_n}^{t_f} T(t)\sigma(\boldsymbol{r}(t))\boldsymbol{c}(\boldsymbol{r}(t), \boldsymbol{v})dt\,, \tag{3}$$

where t_n and t_f are the near and far bounds, respectively, and

$$T(t) = \exp\left(-\int_{t_n}^{t} \sigma(\boldsymbol{r}(s))ds\right)\,. \tag{4}$$

The final rendering result is compared with ground truth color of each ray to optimize the MLP parameter values

$$\mathscr{L} = \sum_{\boldsymbol{r} \in \mathcal{R}} \|C(\boldsymbol{r}) - C_{gt}(\boldsymbol{r})\|_2^2\,, \tag{5}$$

where $\mathcal{R}$ is the ray batch.

4. Method

We aim to realize novel view synthesis as well as refractive surface reconstruction in underwater scenes. We can safely assume that the cameras are placed above the refractive surface. Our method is developed for this forward-facing setup [24], which is also applied to our synthetic and real dataset. Figure 2 shows our pipeline.

Several surface reconstruction approaches[47, 54] opt to learn the surface using a global expression, such as neural signed distance function (SDF). However, the use of this global expression to model our refractive surface will inevitably require intersection computation. Unlike finding intersections with opaque surfaces, finding intersections with refractive surfaces introduces redundant sampling and

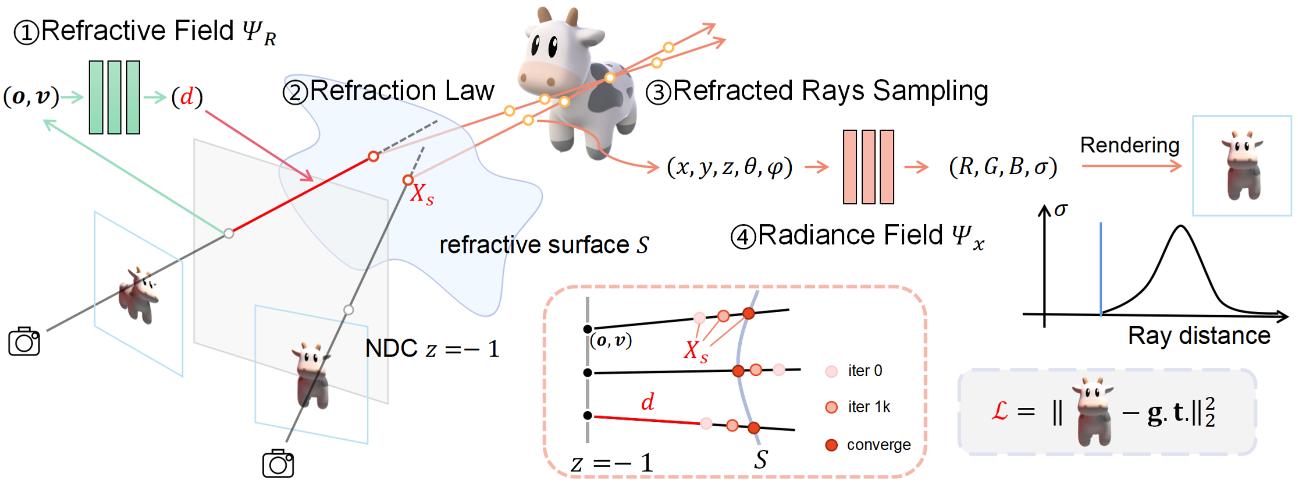


Figure 2: The overall pipeline of NeRFrac. NeRFrac consists of four parts: ① The Refractive Field which estimates the distance from the ray origin to the refractive surface. ② Snell's Law computation which also involves refractive surface normal calculation and coordinate transformations. ③ Sampling on refracted rays. ④ Radiance Field for outputting radiance and density of each spatial point. A simple image reconstruction loss can be used to train NeRFrac.

challenges in convergence. We will detail this analysis in Sec. 6. For clarity, we first describe our approach.

In this work, we model the refractive surface with a neural Refractive Field. Our representation allows direct inference of intersections between rays and the refractive surface, and can maintain multi-view consistency when the network converges. The refractive surface, denoted by $\boldsymbol{S}$, is represented as an open 2-manifold surface. In the subsequent steps, we will illustrate the procedure of underwater novel view rendering and estimating the shape of the refractive surface.

4.1. Refractive Field

Our Refractive Field $\Psi_R(\boldsymbol{o}, \boldsymbol{v}) = d$ is designed to estimate the distance from the input ray origins to the refractive surface. For forward-facing captures, we represent scenes in the normalized device coordinate (NDC). We first calculate the average camera pose from given cameras. Quantities in the average camera coordinate system are denoted with subscript $\boldsymbol{c}$ and otherwise in the NDC frame. First, batches of rays $\boldsymbol{r_c}(t) = \boldsymbol{o_c} + t\boldsymbol{v_c}$ can be calculated given the intrinsic and extrinsic parameters of the cameras, where $\boldsymbol{o_c}$ and $\boldsymbol{v_c}$ stand for the origin and direction vector of each ray, respectively. Then, these rays $\boldsymbol{r_c}$ will be transformed from the average camera coordinate system to the NDC frame, denoted by $\boldsymbol{r}$. After this transformation, every origin $\boldsymbol{o} = (x, y, z)$ should have the same $z = -1$ (*i.e.*, NDC near plane) and every direction vector $\boldsymbol{v} = (v_x, v_y, v_z)$ should follow $v_z > 0$. In NDC, all the ray origins $\boldsymbol{o}$ are arranged on the near plane, and it is easier to interpolate $\boldsymbol{o}$ on a 2D plane than in 3D coordinates.

Origins $\boldsymbol{o}$ and direction vectors $\boldsymbol{v}$ are sent to Ψ_R, which consists of 8 fully-connected layers (using ReLU activations and 256 channels per layer). Ψ_R outputs depth d for each ray. This can be further used to compute $\boldsymbol{X_s}$, the estimated intersection of the input rays and the refractive surface $\boldsymbol{S}$. One ray will intersect the refractive surface at point

$$\boldsymbol{r}(d) = \boldsymbol{o} + d\boldsymbol{v} \,. \tag{6}$$

where d is the Refractive Field output and $\boldsymbol{r}(d)$ is the estimated intersection $\boldsymbol{X_s}$ we mention above.

4.2. Refraction Law in Average Camera Coordinates

Ray $\boldsymbol{r}$ refracts once it arrives at $\boldsymbol{X_s}$. To obtain the refracted ray $\boldsymbol{r'}$, first we need to calculate the normal of the refractive surface at $\boldsymbol{X_s}$. We can then obtain the refracted ray $\boldsymbol{r'}$ based on Snell's Law. Both the vertical relation and the vector form of Snell's Law (see Eq. (2)), however, are derived in the average camera coordinate frame. Since the transform from the average camera coordinate frame to the NDC frame is not Euclidean, we cannot directly calculate the normal and apply Eq. (2) in the NDC frame. An intuitive way is to transform $\boldsymbol{X_s}$ and $\boldsymbol{v}$ (the direction vector of $\boldsymbol{r}$) to average camera coordinates, where the vertical relation and the Snell's Law are valid. After doing this, we have $\boldsymbol{v_c}$ and $\boldsymbol{X_{sc}}$ in the average camera coordinate frame. The next step is the computation of the refractive surface normal and the refracted direction vector $\boldsymbol{v}'_c$ in the average camera coordinate frame.

4.3. Refractive Surface Normal Calculation

To obtain the normal of the refractive surface, first we sample 3×3 neighbor points around $\boldsymbol{X_s}$ in the NDC frame and transform them into the average camera coordinate frame, denoting them as $\boldsymbol{X}_{\text{near}}$. Given 3D coordinates of $\boldsymbol{X}_{\text{near}} = (x_{\text{near}}, y_{\text{near}}, z_{\text{near}})$, we fit a plane using least squares. After centering the coordinates, the loss of this plane fitting becomes

$$J(w) = \frac{1}{2}(P_{xy}w - z_{\text{near}})^{\mathrm{T}}(P_{xy}w - z_{\text{near}})\,, \tag{7}$$

where $P_{xy} = [x_{\text{near}}\ y_{\text{near}}]$ and $w = [w_1\ w_2]^{\mathrm{T}}$. Then we solve this in close-form

$$w^* = ({P_{xy}}^{\mathrm{T}}P_{xy})^{-1}{P_{xy}}^{\mathrm{T}}z_{\text{near}}\,, \tag{8}$$

with QR decomposition. The local surface normal at $\boldsymbol{X_{sc}}$ is given by $\boldsymbol{N_c} = [-w_1, -w_2, 1]$. We next compute the refracted direction vector $\boldsymbol{v_c'}$ according to Eq. (2) and to transform $\boldsymbol{v_c'}$ back to NDC frames, denoted by $\boldsymbol{v'}$. The derivation of this transformation can be found in the supplemental material.

We use neighbor points of X_s to calculate the local surface normal, which takes advantage of the local continuity of our refractive surface. This can safely be assumed for most media including water expect for local perturbations like a splash.

4.4. Sampling and NeRF Rendering Underneath Refractive Surface

The underwater scene area is distributed along the $+z$ direction of $\boldsymbol{S}$. As we approximate $\boldsymbol{S}$ with $\{\boldsymbol{X}_s^i\}_{i=1,2...}$, we can define the ray $\boldsymbol{r'}$ below the refractive surface as

$$\boldsymbol{r'}(t) = \boldsymbol{X}_s + t\boldsymbol{v'}\,. \tag{9}$$

The rays $\boldsymbol{r'}(t)$ lie completely under the refractive surface. For each ray, points $\boldsymbol{x}$ are sampled along $\boldsymbol{r'}(t)$, and these 3D points will be fed to our Radiance Field $\Psi_x(\boldsymbol{x}, \boldsymbol{v'}) = [RGB, \sigma]$ with positional encoding [35] for further prediction of radiance and volume density. We render the scene below the refractive surface according to the path integral Eq. (3) using coarse-to-fine optimization. The rendering result is compared with the ground-truth color, which defines the image loss in Eq. (5).

4.5. Refractive Surface Reconstruction

The refractive surface $\boldsymbol{S}$ consists of a collection of intersection points $\{\boldsymbol{X}_s^i\}_{i=1,2...}$. By visualizing $\boldsymbol{X}_s$, we obtain the shape of our reconstructed refractive surface. We transform $\boldsymbol{X}_s$ to the average camera coordinate frame to visualize it more intuitively.

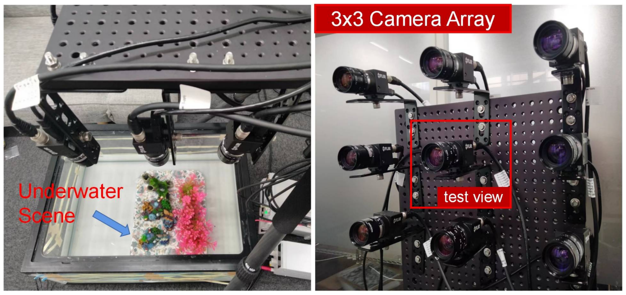


Figure 3: Our setting for capturing real underwater dataset.

Also, note that all rays are below the refractive surface when they are sampled, and points above the refractive surface have no contributions to the final color of rays. That is to say, the Refractive Field Ψ_R encodes information of the refractive surface, and the Radiance Field Ψ_x encodes information about what the scene below the surface truly looks like. This disentanglement allows us to remove Ψ_R in forward inference so that the scene can be viewed as if the observer was in the same medium as it under the refractive surface (*e.g.*, water-free scene restoration). More details and visualizations can be found in Sec. 5.

5. Experimental Results

We evaluate the effectiveness of NeRFrac with an extensive set of experiments on synthetic data and real data in which we focus on water as the refractive surface. We assume the same forward-facing setup as in [24].

5.1. Synthetic Data

To align the real data, our synthetic data are also 3×3 images arranged per scene, generated using ray tracing [20]. The refractive surface is added to get multi-view images of the scene seen through the refractive surface. We use Taichi[18], a programming toolkit for rendering, which can easily and efficiently complete these ray tracing tasks.

We test on two different shapes of common refractive surfaces. Surface A (second sine) and B (primary sine)

$$\begin{aligned} A:&\ \ z = z_0 + a\sin\left(\omega\sqrt{(x-x_0)^2 + (y-y_0)^2}\right), \\ B:&\ \ z = z_0 + a\sin\left(\omega(x+y)\right), \end{aligned} \tag{10}$$

respectively, where a and ω are the amplitude and frequency of the wave, which are configurable parameters.

5.2. Real Data

As shown in Fig. 3, to capture underwater scenes, we build a 3×3 camera array looking down a water surface. Nine cameras (FLIR BFS-U3-23S3C) are synchronized for continuous recording. First, we calibrate the intrinsic and extrinsic parameters of the cameras using COLMAP [39].

	PSNR↑	SSIM↑	LPIPS↓	Average↓
Synthetic data (second sine)				
NeRF[25]	25.93	0.857	0.230	0.061
Mip-NeRF[4]	24.90	0.789	0.381	0.083
IBRNet[48]	26.26	0.857	0.219	0.058
Tensorf[8]	24.60	0.783	0.276	0.076
Plenoxels[56]	17.62	0.340	0.620	0.206
LB-NeRF[13]	26.16	0.873	0.166	0.052
NeRFrac(ours)	**34.98**	**0.948**	**0.142**	**0.022**
Synthetic data (primary sine)				
NeRF[25]	21.01	0.723	0.334	0.112
Mip-NeRF[4]	21.84	0.707	0.448	0.117
IBRNet[48]	21.56	0.728	0.317	0.105
Tensorf[8]	21.20	0.651	0.426	0.124
Plenoxels[56]	17.61	0.325	0.626	0.207
LB-NeRF[13]	22.59	0.810	0.266	0.086
NeRFrac (ours)	**34.38**	**0.945**	**0.148**	**0.023**
Real data (plant)				
NeRF[25]	21.54	0.794	0.235	0.091
Mip-NeRF[4]	25.90	0.728	0.414	0.082
IBRNet[48]	26.89	0.823	0.229	0.058
Tensorf[8]	28.23	0.853	0.178	0.047
Plenoxels[56]	21.72	0.573	0.463	0.127
LB-NeRF[13]	24.52	0.755	0.255	0.076
NeRFrac (ours)	**28.29**	**0.883**	**0.153**	**0.043**
Real data (tree)				
NeRF[25]	28.95	0.804	0.360	0.059
Mip-NeRF[4]	28.91	0.759	0.436	0.065
IBRNet[48]	29.94	0.818	0.268	0.049
Tensorf[8]	29.30	0.805	0.280	0.053
Plenoxels[56]	23.67	0.530	0.533	0.116
LB-NeRF[13]	23.91	0.573	0.357	0.098
NeRFrac (ours)	**31.20**	**0.871**	**0.222**	**0.039**

Table 1: Quantitative comparison of our NeRFrac against other methods. We show the results on both synthetic data and real data. See the text for more details.

Then, objects are placed in a water tank where the depth of the water is unknown. We approximately focus all camera views onto the center of underwater scenes. Our real dataset contains 10 multi-view underwater image sequences, each including the full range of water surfaces from flat to fluctuating. We use our synchronous camera array to take multi-frame images continuously at 5fps. For this experimental part, we train our NeRFrac frame by frame. Yet, our method can be easily adapted to deal with continuous frames as well. We strongly recommend readers to watch the supplemental videos on this extension.

5.3. Comparison

For both synthetic and real data, we use 8 side views for training and the center view for testing (see Fig. 3). We train NeRF [25] with its default configurations for forward-facing data. Then, we consider a set of recent NeRF variants as additional baselines.

Mip-NeRF [4] uses a conical frustum for better sampling and achieves higher accuracy among the NeRF baselines.

IBRNet [48] applies a ray transformer to NeRF MLP estimation. We fine-tune the network with our data.

Tensorf [8] factorizes radiance fields into compact components for scene modeling. This design allows faster and higher-quality rendering.

Plenoxels [56] proposes a sparse voxel model which can be optimized to high fidelity without any neural networks.

LB-NeRF [13] designs a deform layer to directly learn the point offset for refractive scenes rendering and is the pure deformable method we want to compare as stated in Sec. 2. Since it is a very new work with no code available, we try to reproduce this work based on PyTorch [32]. Implementation details can be found in the supplementary material.

Eikonal Fields (EiF) [6] optimizes for a field of 3D-varying IOR and trace light that bends toward the spatial gradients according to the laws of eikonal light transport. The refraction scene it uses, such as a crystal ball, is quite different from our setting, resulting in a bias when using EiF on our data. We show a comparison with this method in the supplementary material.

Qian's Method [34] reconstructs water surface via traditional geometric modeling. Since the source code is not available, in the supplemental material, we present a comparison with this study kindly conducted by the authors of [34] on their data.

We report three error metrics in total, namely the peak signal-to-noise ratio (PSNR), the structural similarity index measure (SSIM) [49], and learned perceptual image patch similarity (LPIPS) [59]. We also use a more intuitive metric, "average" [4], which is the geometric mean of $\mathrm{MSE} = 10^{-\mathrm{PSNR}/10}$, $\sqrt{1 - \mathrm{SSIM}}$, and LPIPS. Fig. 4 and Tab. 1 show the results of NeRFrac and baseline methods.

5.4. 3D Water Surface and Water Removal

Fig. 5 shows the results of water surface reconstruction for both synthesis and real data. Data "second sine" and "primary sine" have waveforms A and B, respectively, as described in Eq. (10). And data "plant," "tree," "red flower," and "fish" are real data that lack ground truth for the water surface. We report the root mean square error (RMSE) to evaluate our water surface reconstruction, which could be found in Tab. 2.

Thanks to the disentanglement in Sec. 4.5, we can easily create views of the underwater scene without the water refraction. For synthetic data, we can render ground truth without water for quantitative evaluation. We use the same metrics as stated in Sec. 5.3 for error estimation. Tab. 2 shows quantitative errors, and Fig. 6 displays the restoration of underwater scenes. As for other benchmarks, we directly remove the deformation fields of LB-NeRF as a reference.

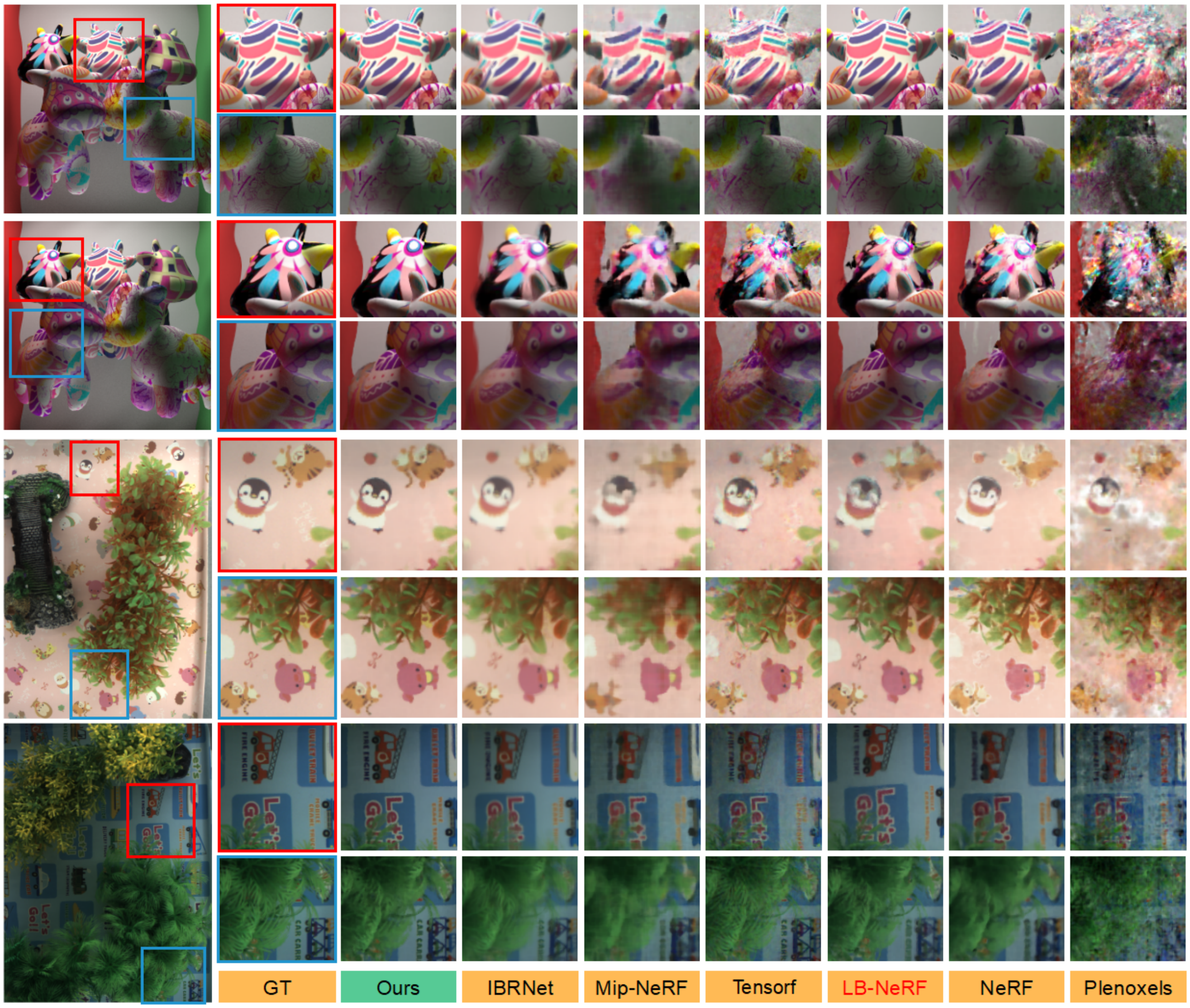


Figure 4: Qualitative Comparison. We show the results of synthetic data: “second sine” and “primary sine,” real data: “plant” and “tree” (from top to bottom). By learning the deformation caused by refraction, our NeRFrac achieves the highest accuracy among all methods (LB-NeRF in red means this work is reproduced by us). More visualizations are shown in the supplemental material.

Notice that the lines shown in the ground-truth water-free images, such as the corner of the wall, should have been straight. They are, however, bent in the ground-truth water-contained images taken from the leave-one-out camera due to refraction. With our method, these lines can be restored to their original shape.

6. 3D Refractive Surface Representations

In this work, we model the refractive surface with a neural Refractive Field, instead of commonly used global representations, such as depth map and SDF. There are two main

	PSNR↑	SSIM↑	LPIPS↓	Average↓	Surface RMSE↓(cm)
Synthetic data (second sine)					
LB-NeRF[13]	16.26	0.499	0.412	0.19	/
NeRFrac (ours)	**29.38**	**0.914**	**0.142**	**0.036**	0.116
Synthetic data (primary sine)					
LB-NeRF[13]	17.26	0.523	0.437	0.178	/
NeRFrac (ours)	**29.94**	**0.918**	**0.144**	**0.035**	0.144

Table 2: Results of virtual water-free scene restoration and water surface reconstruction. We compare our method with LB-NeRF[13] to show our superiority.

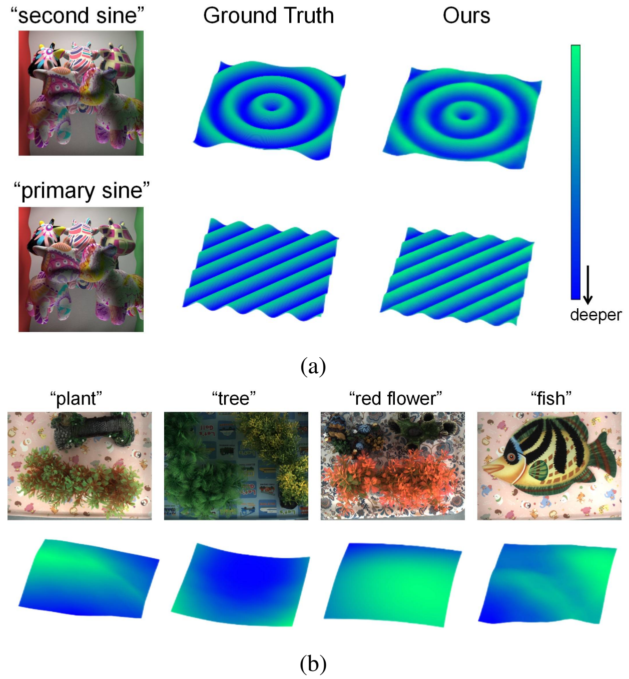


Figure 5: Reconstruction of water surface. The boundaries of these water surfaces are determined by the FOV of the capturing camera. (a) Ground truth and surface reconstruction results of our synthetic data. (b) Surface reconstruction results of our real data (no ground truth available).

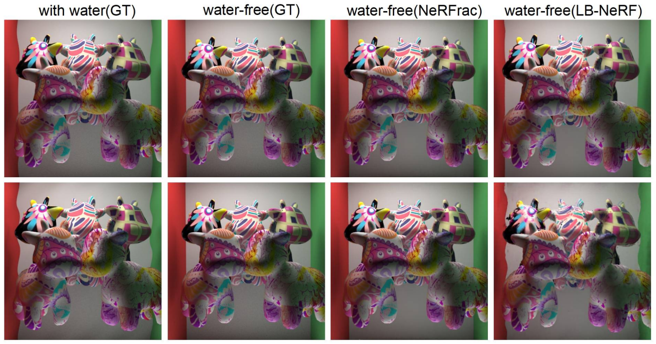


Figure 6: Qualitative result of underwater scene restoration. We can render water-free scenes by removing the Refractive Field. We directly remove the deformation field of LB-NeRF[13] to serve as a comparison. As can be seen, our method successfully restores bent lines thanks to the explicit modeling of Snell's Law, while the purely implicit neural method LB-NeRF fails in restoring water-free images.

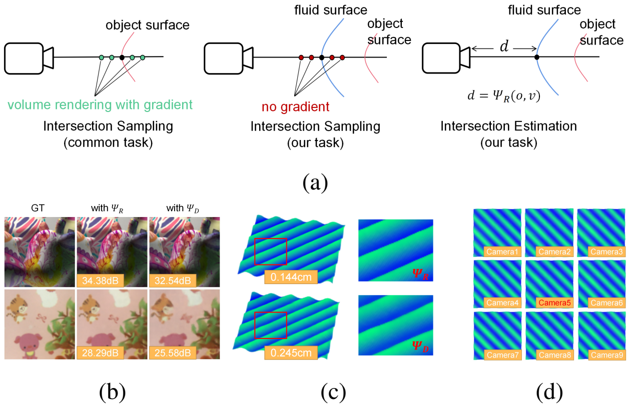


Figure 7: (a) Different intersection calculation strategies. (b)(c) Qualitative and quantitative comparison between Ψ_R and Ψ_D on both novel rendering results and the surface shape (PSNR and RMSE used separately). (d) Multi-view consistency is ensured with our NeRFrac.

considerations.

First, we recall the biggest advantage of having a global representation of the refractive surface: the ability to calculate the surface normal by differentiating the surface thanks to its spatial continuity. This advantage, however, cannot be exploited in our scenario. We learn the refractive surface in NDC frames, where the gradient direction is no longer in line with the refractive surface normal direction. As a result, differentiation for computing the refractive surface normal becomes challenging.

Second, a globally defined surface representation inevitably necessitates extra sampling when calculating the intersection between rays and the surface. As can be seen in Fig. 7 (a), for common opaque surface reconstruction, these sampled points will get the gradient from volume rendering to update the surface shape. For our refractive surface reconstruction, however, we do not use these sampled points for volume rendering, so most of them will fail to receive the gradient, making surface shape update a challenge.

For further comparison, we replace our Refractive Field Ψ_R with a self-implemented global depth map network Ψ_D, which requires sampling for intersection calculation. The design details of Ψ_D can be found in the supplementary material. In Fig. 7 (b)(c) we show the results of Ψ_D compared with Ψ_R. The challenge in gradient back propagation of Ψ_D adds blurs and artifacts to rendered image as well as the reconstructed refractive surface.

Our representation successfully avoids explicit intersection sampling and its consequent hurdles, without sacrificing multi-view consistency. Note that, as we parameterize the ray direction vectors $\boldsymbol{v}$ as the input of our Refractive Field Ψ_R, this implicitly serves as a prior for promoting multi-view consistency. As experimentally demonstrated in Fig. 7 (d), when the network converges, the multi-view consistency is guaranteed and $\boldsymbol{X}_s$ will converges to the NDC global manifold S.

7. Conclusion

We presented NeRFrac to tackle novel view synthesis through refractive surfaces by introducing an MLP-based Refractive Field. Unlike NeRF and most other related works which sample directly on straight lines, we design a Refractive Field to implicitly encode the refractive surface, from which we calculate the refracted rays based on Snell's Law. Training the same epochs at a speed similar to NeRF, we can achieve better results than advanced variants of NeRF. NeRFrac can also reconstruct the refractive surface and create refraction-free scenes by removing the learned Refractive Field. NeRFrac is currently limited to modeling a single layer of refraction. We plan to extend the method to handle complex layered refraction as well as other light behavior including scattering and caustic in our future work.

Acknowledgements

This research was supported in part by JSPS KAKENHI Grant Numbers 22H00529, 21H04893, 20H05951.

References

[1] Marina Alterman, Yoav Y Schechner, and Yohay Swirski. Triangulation in Random Refractive Distortions. *IEEE transactions on pattern analysis and machine intelligence*, 39(3):603–616, 2016. 3

[2] ShahRukh Athar, Zexiang Xu, Kalyan Sunkavalli, Eli Shechtman, and Zhixin Shu. RigNeRF: Fully Controllable Neural 3D Portraits. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 20364–20373, 2022. 2

[3] Benjamin Attal, Jia-Bin Huang, Michael Zollhöfer, Johannes Kopf, and Changil Kim. Learning Neural Light Fields With Ray-Space Embedding. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 19819–19829, 2022. 2

[4] Jonathan T Barron, Ben Mildenhall, Matthew Tancik, Peter Hedman, Ricardo Martin-Brualla, and Pratul P Srinivasan. Mip-nerf: A Multiscale Representation for Anti-aliasing Neural Radiance Fields. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 5855–5864, 2021. 2, 6

[5] Jonathan T Barron, Ben Mildenhall, Dor Verbin, Pratul P Srinivasan, and Peter Hedman. Mip-nerf 360: Unbounded Anti-aliased Neural Radiance Fields. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 5470–5479, 2022. 2

[6] Mojtaba Bemana, Karol Myszkowski, Jeppe Revall Frisvad, Hans-Peter Seidel, and Tobias Ritschel. Eikonal Fields for Refractive Novel-View Synthesis. In *ACM SIGGRAPH 2022 Conference Proceedings*, pages 1–9, 2022. 2, 6

[7] Eric R Chan, Connor Z Lin, Matthew A Chan, Koki Nagano, Boxiao Pan, Shalini De Mello, Orazio Gallo, Leonidas J Guibas, Jonathan Tremblay, Sameh Khamis, et al. Efficient Geometry-aware 3D Generative Adversarial Networks. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 16123–16133, 2022. 2

[8] Anpei Chen, Zexiang Xu, Andreas Geiger, Jingyi Yu, and Hao Su. Tensorf: Tensorial radiance fields. In *Computer Vision–ECCV 2022: 17th European Conference, Tel Aviv, Israel, October 23–27, 2022, Proceedings, Part XXXII*, pages 333–350. Springer, 2022. 2, 6

[9] Tianlong Chen, Peihao Wang, Zhiwen Fan, and Zhangyang Wang. Aug-NeRF: Training Stronger Neural Radiance Fields With Triple-Level Physically-Grounded Augmentations. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 15191–15202, 2022. 2

[10] Kangle Deng, Andrew Liu, Jun-Yan Zhu, and Deva Ramanan. Depth-supervised NeRF: Fewer Views and Faster Training for Free. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 12882–12891, 2022. 2

[11] Yu Deng, Jiaolong Yang, Jianfeng Xiang, and Xin Tong. Gram: Generative Radiance Manifolds for 3d-aware Image Generation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 10673–10683, 2022. 2

[12] Ricardo Ferreira, Joao P Costeira, and Joao A Santos. Stereo Reconstruction of a Submerged Scene. In *Iberian Conference on Pattern Recognition and Image Analysis*, pages 102–109. Springer, 2005. 3

[13] Taku Fujitomi, Ken Sakurada, Ryuhei Hamaguchi, Hidehiko Shishido, Masaki Onishi, and Yoshinari Kameda. LB-NERF: Light Bending Neural Radiance Fields for Transparent Medium. In *2022 IEEE International Conference on Image Processing (ICIP)*, pages 2142–2146. IEEE, 2022. 2, 6, 7, 8

[14] Stephan J Garbin, Marek Kowalski, Matthew Johnson, Jamie Shotton, and Julien Valentin. Fastnerf: High-fidelity Neural Rendering at 200fps. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 14346–14355, 2021. 2

[15] Yuan-Chen Guo, Di Kang, Linchao Bao, Yu He, and Song-Hai Zhang. Nerfren: Neural Radiance Fields with Reflections. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 18409–18418, 2022. 2

[16] Yang Hong, Bo Peng, Haiyao Xiao, Ligang Liu, and Juyong Zhang. Headnerf: A Real-time Nerf-based Parametric Head Model. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 20374–20384, 2022. 2

[17] Tao Hu, Shu Liu, Yilun Chen, Tiancheng Shen, and Jiaya Jia. EfficientNeRF: Efficient Neural Radiance Fields. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 12902–12911, 2022. 2

[18] Yuanming Hu, Tzu-Mao Li, Luke Anderson, Jonathan Ragan-Kelley, and Frédo Durand. Taichi: A Language for High-performance Computation on Spatially Sparse Data Structures. *ACM Transactions on Graphics (TOG)*, 38(6):1–16, 2019. 5

[19] James T Kajiya and Brian P Von Herzen. Ray Tracing Volume Densities. *ACM SIGGRAPH computer graphics*, 18(3):165–174, 1984. 3

[20] Timothy L Kay and James T Kajiya. Ray Tracing Complex Scenes. *ACM SIGGRAPH computer graphics*, 20(4):269–278, 1986. 2, 5

[21] Chuan Li, Martin Shaw, David Pickup, Darren Cosker, Phil Willis, and Peter Hall. Realtime Video Based Water Surface Approximation. In *2011 Conference for Visual Media Production*, pages 109–117. IEEE, 2011. 3

[22] Lingjie Liu, Jiatao Gu, Kyaw Zaw Lin, Tat-Seng Chua, and Christian Theobalt. Neural Sparse Voxel Fields. *Advances in Neural Information Processing Systems*, 33:15651–15663, 2020. 2

[23] Li Ma, Xiaoyu Li, Jing Liao, Qi Zhang, Xuan Wang, Jue Wang, and Pedro V Sander. Deblur-NeRF: Neural Radiance Fields from Blurry Images. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 12861–12870, 2022. 2

[24] Ben Mildenhall, Pratul P Srinivasan, Rodrigo Ortiz-Cayon, Nima Khademi Kalantari, Ravi Ramamoorthi, Ren Ng, and Abhishek Kar. Local Light Field Fusion: Practical View Synthesis with Prescriptive Sampling Guidelines. *ACM Transactions on Graphics (TOG)*, 38(4):1–14, 2019. 3, 5

[25] Ben Mildenhall, Pratul P Srinivasan, Matthew Tancik, Jonathan T Barron, Ravi Ramamoorthi, and Ren Ng. Nerf: Representing Scenes as Neural Radiance Fields for View Synthesis. *Communications of the ACM*, 65(1):99–106, 2021. 1, 2, 3, 6

[26] Nigel JW Morris and Kiriakos N Kutulakos. Dynamic Refraction Stereo. *IEEE transactions on pattern analysis and machine intelligence*, 33(8):1518–1531, 2011. 3

[27] Xiaoying Nie, Yong Hu, and Xukun Shen. Physics-preserving Fluid Reconstruction from Monocular Video Coupling with SFS and SPH. *The Visual Computer*, 36(6):1247–1257, 2020. 3

[28] Michael Niemeyer, Jonathan T Barron, Ben Mildenhall, Mehdi SM Sajjadi, Andreas Geiger, and Noha Radwan. Regnerf: Regularizing Neural Radiance Fields for View Synthesis from Sparse Inputs. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 5480–5490, 2022. 2

[29] Michael Oechsle, Songyou Peng, and Andreas Geiger. Unisurf: Unifying Neural Implicit Surfaces and Radiance Fields for Multi-view Reconstruction. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 5589–5599, 2021. 2

[30] Roy Or-El, Xuan Luo, Mengyi Shan, Eli Shechtman, Jeong Joon Park, and Ira Kemelmacher-Shlizerman. Stylesdf: High-resolution 3d-consistent Image and Geometry Generation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 13503–13513, 2022. 2

[31] Keunhong Park, Utkarsh Sinha, Jonathan T Barron, Sofien Bouaziz, Dan B Goldman, Steven M Seitz, and Ricardo Martin-Brualla. Nerfies: Deformable Neural Radiance Fields. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 5865–5874, 2021. 2

[32] Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, et al. Pytorch: An Imperative Style, High-performance Deep Learning Library. *Advances in neural information processing systems*, 32, 2019. 6

[33] Albert Pumarola, Enric Corona, Gerard Pons-Moll, and Francesc Moreno-Noguer. D-nerf: Neural Radiance Fields for Dynamic Scenes. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 10318–10327, 2021. 2

[34] Yiming Qian, Yinqiang Zheng, Minglun Gong, and Yee-Hong Yang. Simultaneous 3d Reconstruction for Water Surface and Underwater Scene. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 754–770, 2018. 3, 6

[35] Nasim Rahaman, Aristide Baratin, Devansh Arpit, Felix Draxler, Min Lin, Fred Hamprecht, Yoshua Bengio, and Aaron Courville. On the Spectral Bias of Neural Networks. In *International Conference on Machine Learning*, pages 5301–5310. PMLR, 2019. 5

[36] Daniel Rebain, Mark Matthews, Kwang Moo Yi, Dmitry Lagun, and Andrea Tagliasacchi. LOLNeRF: Learn from One Look. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 1558–1567, 2022. 2

[37] Christian Reiser, Songyou Peng, Yiyi Liao, and Andreas Geiger. Kilonerf: Speeding Up Neural Radiance Fields with Thousands of Tiny Mlps. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 14335–14345, 2021. 2

[38] Barbara Roessle, Jonathan T Barron, Ben Mildenhall, Pratul P Srinivasan, and Matthias Nießner. Dense Depth Priors for Neural Radiance Fields from Sparse Input Views. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 12892–12901, 2022. 2

[39] Johannes L Schonberger and Jan-Michael Frahm. Structure-from-motion Revisited. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 4104–4113, 2016. 5

[40] Advaith Venkatramanan Sethuraman, Manikandasriram Srinivasan Ramanagopal, and Katherine A Skinner. Waternerf: Neural radiance fields for underwater scenes. *arXiv preprint arXiv:2209.13091*, 2022. 2

[41] Mohammed Suhail, Carlos Esteves, Leonid Sigal, and Ameesh Makadia. Light Field Neural Rendering. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 8269–8279, 2022. 2

[42] Cheng Sun, Min Sun, and Hwann-Tzong Chen. Direct Voxel Grid Optimization: Super-fast Convergence for Radiance Fields Reconstruction. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 5459–5469, 2022. 2

[43] Jingxiang Sun, Xuan Wang, Yong Zhang, Xiaoyu Li, Qi Zhang, Yebin Liu, and Jue Wang. Fenerf: Face Editing in Neural Radiance Fields. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 7672–7682, 2022. 2

[44] Simron Thapa, Nianyi Li, and Jinwei Ye. Dynamic Fluid Surface Reconstruction Using Deep Neural Network. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 21–30, 2020. 3

[45] Dor Verbin, Peter Hedman, Ben Mildenhall, Todd Zickler, Jonathan T Barron, and Pratul P Srinivasan. Ref-nerf: Structured View-dependent Appearance for Neural Radiance Fields. *arXiv preprint arXiv:2112.03907*, 2021. 1, 2

[46] Liao Wang, Jiakai Zhang, Xinhang Liu, Fuqiang Zhao, Yanshun Zhang, Yingliang Zhang, Minye Wu, Jingyi Yu, and Lan Xu. Fourier PlenOctrees for Dynamic Radiance Field Rendering in Real-time. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 13524–13534, 2022. 2

[47] Peng Wang, Lingjie Liu, Yuan Liu, Christian Theobalt, Taku Komura, and Wenping Wang. Neus: Learning neural implicit surfaces by volume rendering for multi-view reconstruction. *arXiv preprint arXiv:2106.10689*, 2021. 2, 3

[48] Qianqian Wang, Zhicheng Wang, Kyle Genova, Pratul P Srinivasan, Howard Zhou, Jonathan T Barron, Ricardo Martin-Brualla, Noah Snavely, and Thomas Funkhouser. Ibrnet: Learning Multi-view Image-based Rendering. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 4690–4699, 2021. 2, 6

[49] Zhou Wang, Alan C Bovik, Hamid R Sheikh, and Eero P Simoncelli. Image Quality Assessment: From Error Visibility to Structural Similarity. *IEEE transactions on image processing*, 13(4):600–612, 2004. 6

[50] Ziyu Wang, Wei Yang, Junming Cao, Lan Xu, Junqing Yu, and Jingyi Yu. NeReF: Neural Refractive Field for Fluid Surface Reconstruction and Implicit Representation. *arXiv preprint arXiv:2203.04130*, 2022. 3

[51] Yi Wei, Shaohui Liu, Yongming Rao, Wang Zhao, Jiwen Lu, and Jie Zhou. Nerfingmvs: Guided Optimization of Neural Radiance Fields for Indoor Multi-view Stereo. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 5610–5619, 2021. 2

[52] Jinhui Xiong and Wolfgang Heidrich. In-the-wild Single Camera 3D Reconstruction through Moving Water Surfaces. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 12558–12567, 2021. 3

[53] Qiangeng Xu, Zexiang Xu, Julien Philip, Sai Bi, Zhixin Shu, Kalyan Sunkavalli, and Ulrich Neumann. Point-nerf: Point-based Neural Radiance Fields. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 5438–5448, 2022. 2

[54] Lior Yariv, Jiatao Gu, Yoni Kasten, and Yaron Lipman. Volume rendering of neural implicit surfaces. *Advances in Neural Information Processing Systems*, 34:4805–4815, 2021. 2, 3

[55] Lior Yariv, Yoni Kasten, Dror Moran, Meirav Galun, Matan Atzmon, Basri Ronen, and Yaron Lipman. Multiview neural surface reconstruction by disentangling geometry and appearance. *Advances in Neural Information Processing Systems*, 33:2492–2502, 2020. 3

[56] Alex Yu, Sara Fridovich-Keil, Matthew Tancik, Qinhong Chen, Benjamin Recht, and Angjoo Kanazawa. Plenoxels: Radiance Fields without Neural Networks. *arXiv preprint arXiv:2112.05131*, 2021. 2, 6

[57] Alex Yu, Ruilong Li, Matthew Tancik, Hao Li, Ren Ng, and Angjoo Kanazawa. Plenoctrees for Real-time Rendering of Neural Radiance Fields. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 5752–5761, 2021. 2

[58] Mingjie Zhang, Xing Lin, Mohit Gupta, Jinli Suo, and Qionghai Dai. Recovering Scene Geometry under Wavy Fluid via Distortion and Defocus Analysis. In *European Conference on Computer Vision*, pages 234–250. Springer, 2014. 3

[59] Richard Zhang, Phillip Isola, Alexei A Efros, Eli Shechtman, and Oliver Wang. The Unreasonable Effectiveness of Deep Features as a Perceptual Metric. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 586–595, 2018. 6